Energy Demand Prediction: A Partial Information Game Approach

Hélène Le Cadre, Roman Potarusov, Cédric Auliac

Energy Demand Prediction: A Partial Information Game Approach

Hélène Le Cadre* Roman Potarusov† Cédric Auliac‡

September 22, 2011

Abstract

This article proposes an original approach to predict the electric vehicles (EVs)' energy demand in a charge station using a regret minimization learning approach. The problem is modelled as a two players game involving: on the one hand the EV drivers, whose demand is unknown and, on the other hand, the service provider who owns the charge station and wants to make the best predictions in order to minimize his regret. The information in the game is partial. Indeed, the service provider never observes the EV drivers' energy demand. The only information he has access to is contained in a feedback function which depends on his predictions accuracy and on the EV drivers' consumption level. The local/expanded accuracy and the ability for uncertainty handling of the regret minimization learning approach is evaluated by comparison with three well-known learning approaches: (i) Neural Network, (ii) Support Vector Machine, (iii) AutoRegressive Integrated Moving Average process, using as benchmarks two data bases: an artificial one generated using a bayesian network and real domestic household electricity consumption data in southern California. We observe that over real data, regret minimization algorithms clearly outperform the other learning approaches. The efficiency of these methods open the door to a wide class of game theory applications dealing with collaborative learning, information sharing and manipulation.

Keywords: Partial Information; Repeated Game with Signals; Regret; Non Linear Time Series

1 Introduction

To reduce the CO_2 pollution level and ensure their energetic independence while providing new sources of economic growth, governments plan to invest in the deployment of electric vehicles (EVs) on the market in a very near future [19]. However, EV launching poses many questions on the kind of charging infrastructure to be deployed i.e., either privilege a private usage with charge spots at home or, on the contrary, invest massively so as to develop a dense network of public charge stations[1].

Under the assumption that a dense network of public charge stations were deployed, a major difficulty would be to evaluate the availability of the charging infrastructure. It implies that the ability to develop services associated with the charging infrastructure management, such as providing information to both the

*CEA, LIST, Laboratory Information, Models and Learning, 91191 Gif-sur-Yvette CEDEX, FRANCE. *E-mail*: helene.le-cadre@cea.fr

†LAMIH, university of Valenciennes, Valenciennes, FRANCE. *E-mail*: potarusov.roman@gmail.com

‡CEA, LIST, Laboratory Information, Models and Learning, 91191 Gif-sur-Yvette CEDEX, FRANCE. *E-mail*: cedric.auliac@cea.fr

[1]Note that a charge station is made of a certain number of charge spots to which EV drivers can connect in order to reload their cars.

EV drivers and the providers, becomes strategic to contribute to the EV adoption, by avoiding the drivers' range anxiety. But, the development of these services is conditional to the capacity to access and manage online information regarding resource availability and predicted demand evolution. Practically, there are two levels of requirements concerning energy demand prediction: providers expect short-term forecasts to optimize their resource access and revenue, whereas the EV drivers wait for real-time ones since their decision to reload in a specific charge station rely on the accessible online information concerning the charge station availability. Note that the aim of this article is not to provide an embedded software since we do not impose constraints over criticity, reactivity, autonomy or security.

Besides, underlying uncertainties regarding the EV drivers' spatio-temporal behaviors as well as strong disparity of behaviors over a wide geographic area add another level of complexity to our prediction model. In this article, we have chosen to focus exclusively on the temporal aspect by restricting our predictions to a limited geographic area. Furthermore, we choose to focus on charge stations offering services based on regular/quick charges. In practice, regular charges last between 6 hours and 8 hours whereas quick charges last around 30 minutes depending on the type of battery used [12]. Consequently, as a first approach, it appears quite reasonable to make predictions with a daily time-scale. Note that under the assumption that every day demand is generated according to a known density function with unknown parameter, it is possible to obtain finer granularity predictions over each day, by generating predictions with a finer time-scale, according to the estimated density.

Another striking difficulty is that in general, demand does not coincide with consumption. Demand might be monitored by the service provider by aggregating at a specific time instant, the calls of the EV drivers who declare to need to reload in the charge station, at that specific time instant. However, many reasons can incite the EV drivers to change their minds: unexpected events (like the will to avoid traffic jams, plan changes on the agenda, the borrowing of an alternative route, etc.), the fact that the charge station is already full i.e., blocked, too long mean waiting times, etc. Therefore, to try to get a good estimate of the true consumption level and provide realistic availability measures (mean waiting time, blocking probability) in the charge station, we represent the charge station as a queue [2] with a finite number of parallell servers (charge spots) and a finite length buffer (the car parks). The use of a queue might be of particular interest in further applications dealing with the charge station buffer/spot sizing. Demand is supposed stationary over a day and we assume that it is generated according to a Poisson process whose parameter varies daily [2], [26]. The challenge then, is to develop efficient learning algorithms enabling us to predict the average demand value over the day to come, using past information. Four learning approaches are compared: (i) Neural Network, (ii) Support Vector Machine, (iii) AutoRegressive Integrated Moving Average process, (iv) Regret minimization algorithms. Their performances are compared on local/expanded accuracy, the ability of uncertainty handling, using as benchmarks two data bases: an artifical one generated using a bayesian network and real domestic household electricity consumption data in southern California. Finally, the conclusions issued from these comparisons allow us to provide guidelines to the development of future energy based management models, scalable enough to handle uncertainty/partial information and competition/cooperation.

The article is organized as follows. We start by describing the general prediction problem in Section 2. Then, the data bases used as input of the learning methods tested in this article, are described in Section 3. Their respective information qualities are compared to get a first insight of the difficulty for the learning process to provide accurate forecasts. Measures to compare the learning approach performances are introduced in Subsection 3.4. Taking a system point of view, the problem is modelled as a partial information game in Section 4. It is then solved using learning algorithms based on regret minimization in Subsection 4.3 and the convergence of the algorithms, guaranteeing that the learning process has reached an equilibrium,

is observed over both data bases. Finally, the performances of the regret minimization learning method is compared with classical learning approaches based on non linear time series, in Section 5. We conclude in Section 6.

Note that paragraphs that are written in orange, can be omitted in a first lecture.

2 Problem description

We make the initial assumption that demand is generated on each day j according to a *parametric density function* whose parameter $\lambda(j)$ is unknown[2]. This parameter belongs to a closed subset of the real numbers i.e., $\Lambda \subseteq \mathbb{R}$.

We aim at validating the originality and goodness of fit of regret minimization learning approach, as applied to our energy demand prediction problem. To reach this goal, we need to compare the performances resulting from this learning method, to the ones obtained using more classical learning approaches such as: (i) Neural Network (NN), (ii) Support Vector Machine (SVM), (iii) AutoRegressive Integrated Moving Average process (ARIMA). Explicit measures of performance for the considered learning approaches are introduced in Section 3.4. The learning methods are run over two data bases: an artificial one generated using a bayesian network and a real data base containing household electricity consumption in south California over year 2011. Both data bases are divided into two sets: a training set over which learning is performed and a test set over which simulation is performed.

The classical learning approaches such as NN, SVM, ARIMA are used as non linear time series. We give a short explanation of how they work (cf. Figure 1). The data in the training set are sorted in a triplet containing the day, the month and the associated demand parameter $\lambda(.)$. As a result, it is possible to estimate the demand parameter as a function of day and month. Then, simulation is performed over the test set using as input the values of days and months. It gives as output estimated demand parameters $\hat{\lambda}(.)$. Finally, these estimates can be compared to the true ones in order to evalute the method performances.

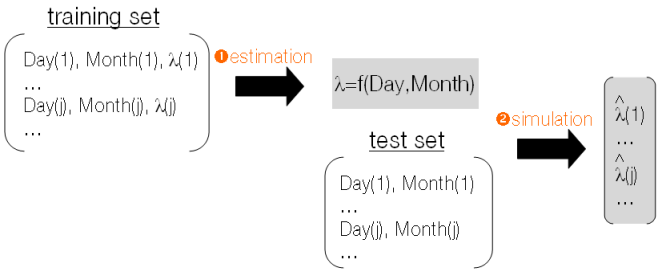

Figure 1: Learning based on non linear time series.

The purpose of regret minimization learning is quite different (cf. Figure 2). Indeed, it does not aim at providing punctual estimates of the demand parameters but rather at providing an estimate of the discrete density function over the set containing all the possible demand parameter values[3]. It works as follows: for

[2]Note that in all generalities, this parameter might be scalar or vectorial.

[3]With hierarchical approaches using NN or SVM, it might also be possible to predict the parametric parameter generating density parameter. However, none of these approaches enable a true incorporation of a strategic rival process playing against the forecaster, like a competitor, in the model.

each hidden[4] value in the training set, the forecaster who coincides with the service provider, tries to build a discrete density function. He then generates an estimate of the demand parameter according to this discrete density function. The forecaster has never access to the true demand parameter but receives a feedback $h(\lambda(.), \hat{\lambda}(.))$ which depends on the hidden demand parameter and on the estimate that he has generated. The feedback function is the observation also called signal in game theory, that the forecaster receives. It enables the forecaster to refine his knowledge about the demand generating density function. Once all the training set has been covered, the resulting estimated density function is used over the test set. Practically, we pick sequentially demand parameter in the test set, compare it to all possible values of λ in Λ and integrate these differences using the estimated density function as the integration measure. In this way, it is possible to compare regret minimization learning performances with the ones obtained using non linear time series.

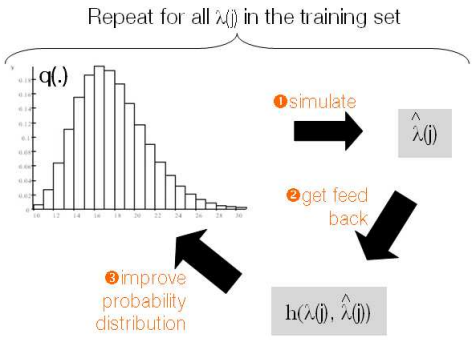

Figure 2: Learning based on regret minimization.

3 Description of the experimental sets

3.1 Artificial data generation

Bayesian networks also known as belief networks, belong to the family of probabilistic graphical models. One of their specific interest is that they enable an effective representation and computation of the joint probability distribution over a set of random variables. They are traditionally used to represent knowledge about an uncertain domain. In particular, each node in the graph represents a random variable while the edges between the nodes represent probabilistic dependencies among the corresponding variables. The weights in the graph are estimated by using known statistical methods that we describe next.

The structure of the network is defined by two sets: the set of nodes and the set of directed edges. It reflects a simple independence statement: each variable is independent ot its non-descendents in the graph given the state of its parents. This property is used to reduce, sometimes significantly, the number of parameters that are required to characterize the joint probability distribution of the variables. This reduction provides an efficient way to compute the posterior probabilities given the evidence.

In addition to the graph structure, one needs to specifiy the parameters of the model. The parameters are described in a manner which is consistent with a Markovian property [20], where the conditional probability distribution at each node depends only on its parents. For discrete random variables, this conditional

[4]The value is hidden because it is not revealed to the forecaster.

probability is often represented by a table, listing the local probability that a child node takes on each of its feasible values. The joint distribution of a collection of variables can be determined uniquely by these local conditional probability tables.

We now describe the static bayesian network that we consider. It is pictured in Figure 3. At time instant t, EV drivers' electricity demand represented by the variable *Energy Demand* (denoted D) depends on the traffic congestion level represented by the variable *Traffic* (denoted T) and on the weather represented by the variable *Weather* (denoted W). Indeed, a high level of congestion might increase the probability that EV drivers be trapped in traffic jams forcing them to adopt irregular speeds for longer periods, which in turn, might increase their electricity consumption. Identically, extreme weather conditions like snow, fog, heavy rains or heavy waves, might force the EV drivers to consume more electricity by adapting their speed, switching on heat or air conditioning, putting the headlights or wipers. The *Traffic* variable depends in turn on the hour variable represented by the variable *Hour* (denoted H) and on the working status variable represented by the variable *Working Statut* (denoted WS). Indeed, the traffic congestion level relies heavily on the considered hour of the day but might adopt a quite different profile depending on whether the day is worked or coincides with holiday. Finally, the variable *Working Statut* depends on the variable *Day* (denoted J) and on the variable *Month* (denoted by M) which points also in the direction of the variable *Weather*.

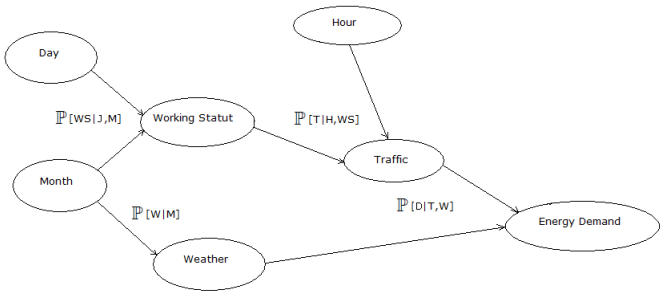

Figure 3: Description of the graph of the variables.

The 6 variables: *Hour*, *Day*, *Month*, *Working Statut*, *Weather*, *Traffic* are supposed discrete. We give in Table 1, the quantified feasible values of the underlying graph explicative variables. We make some simplifying assumptions: a day is divided in slices of 4 hours, each day of a week is associated with an integer number in interval $[\![1;7]\!]$; each month of a year is associated with an integer number in interval $[\![1;12]\!]$. To give some explanations, day 1 coincides with monday and month 1 with january. A day is either worked corresponding to state 1 or coincides with holiday corresponding to state 0. The weather is classified in 5 states. Some correlations may exist between some of them: for example, a day might be both rainy and sunny or sunny and stormy.

We now describe the parameters of the bayesian network. Their computation requires access to data bases. To compute $\mathbb{P}[W|M]$, we have used a data base provided by Meteo-France, for the town of Agen, located in the south-west of France, during year 2011[5]. To obtain $\mathbb{P}[WS|J,M]$, we have listed the holiday days corresponding to specific celebrations or commemorations for year 2011 in France[6], as well as the school holidays for year 2011[7] conditionally to the fact that solely 68% of the France inhabitants will take

[5]The data base is available at: http://www.meteorologic.net/.
[6]The complete list is available at: http://www.calendrier-2011.fr/jours-feries.php/year=2011.
[7]The schedule of the french holidays for year 2011 is available at: http://www.calendrier-2011.fr/.

Hour	0, 4, 8, 12, 16, 20
Day	⟦1;7 ⟧
Month	⟦1;12 ⟧
Working Statut	0, 1
Weather	rain, snow, sun, storm, fog
Traffic	green, orange, red, black

Table 1: Quantified value of underlying graph discrete variables.

some holidays this summer[8]. To compute $\mathbb{P}[T|H, WS]$, we have used both the number of days taking values green, orange, red or black over a year on a french online forecasting website Bison Futé[9] and discretized the congestion density function per day as provided by traffic congestion forecaster website Sytadin[10], over 4 hour lasting slices. Finally, we have assumed that the demand function is distributed according to a gaussian density function whose parameters i.e., mean value and standard deviation, vary as functions of variables T and W values. For the sake of realism, we make the assumptions that the increase of the traffic congestion implies the decrease of the mean value and standard deviation toward zero. Furthermore, the standard deviation is smaller under snow and sun than under fog, rain and storm. These assumptions translate the fact that under extreme conditions, the demand becomes immediately very high i.e., there is a peak of load, since every EV drivers needs simultaneously a high level of electricity. Note that the use of gaussian density function to approximate demand distribution over the lead time interval are quite common in the operations research community [7], [29], to tackle stock-control problems [7], to solve bandwidth provisioning and route selection problems in networks [16], etc.

To highlight the practical interest of the bayesian network, it is worth mentioning that the conditional independence statement of the bayesian network provides a compact factorization of the joint probability distribution. Instead of factorizing the joint probability distribution of all the variables by the chain rule [20] like as follows:

$$\begin{aligned}\mathbb{P}[J, M, WS, W, H, T, D] &= \mathbb{P}[J]\mathbb{P}[M|J]\mathbb{P}[WS|J, M]\mathbb{P}[W|J, M, WS]\mathbb{P}[H|J, M, WS, W] \\ &\quad \mathbb{P}[T|J, M, WS, W, H]\mathbb{P}[D|J, M, WS, W, H, T]\end{aligned}$$

the bayesian network defines a unique joint probability distribution in a factored form:

$$\mathbb{P}[J, M, WS, W, H, T, D] = \mathbb{P}[J]\mathbb{P}[M]\mathbb{P}[WS|J, M]\mathbb{P}[W|M]\mathbb{P}[H]\mathbb{P}[T|H, WS]\mathbb{P}[D|T, W]$$

To obtain dynamic energy demand, we iteratively run the bayesian network to generate a large sample of energy demand values depending on day and month. When day and month associated with a specific energy demand value coincide, we realize an average of the simulated values.

3.2 Southern California household electricity demand data base

The second data base contains the household electric consumption per hour in southern California, from january 2011 to august 2011, based on real data[11]. The values are recorded in kWh. Additionally, statements

[8] Information available in the news forum: http://www.rmc.fr/forum/rmc/france/.

[9] http://www.bisonfute.equipement.gouv.fr/fr/IMG/pdf/

[10] http://www.sytadin.fr/

[11] This data base is reported by SCE: http://www.sce.com/.

were made over residential, small and medium commercial/industrial customers. In Figure 4, we have plotted the average daily demand, during the first 9 months of year 2011. We observe that the data base contains extreme events like in june-july 2011, where electricity consumption peaks appear. These load peaks might be due to severe heat waves, which are rather frequent in southern California.

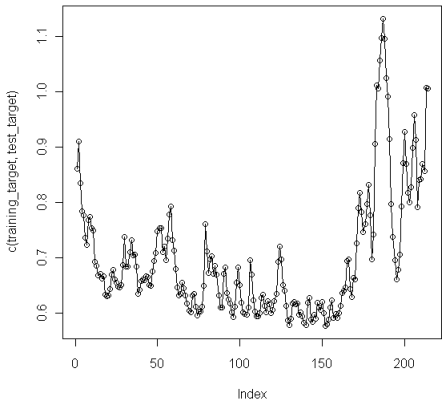

Figure 4: Average daily household electricity demand for year 2011 in southern California ($01/01/2011$-$08/02/2011$).

3.3 Comparison of the information quality in both data bases

In this section, we characterize the quality of the information contained in the data bases used as inputs of our learning algorithms. This step enables us to identify whether the bases are fully informative or only slightly, which might increase the difficulty to build efficient predictors.

The training set extends over $\bar{d}$ days with n samples per day. We let $D(j,t_l),\ j=1,...,\bar{d},\ l=1,...,n$ be the demand over day j at time instant t_l. In all the article, we make the assumption that the electricity demand $D(j,t_l)$ is distributed according to a Poisson density function of parameter $\lambda(j)$ over day $j=1,...,\bar{d}$, at any time instant $l=1,...,n$. We recall that a random variable $D(j,t_l)$ is said to be distributed according to a Poisson density function of parameter $\lambda(j)$ i.e., $D(j,t_l)\sim\mathcal{P}(\lambda(j))$, if, and only if, $\mathbb{P}[D(j,t_l)=k]=\exp(-\lambda(j))\frac{\lambda(j)^k}{k!}$. This assumption will be justified with more details in Section 4.1, where the charge station is identified with a queue. The choice of the Poisson density function simplifies widely the estimation of its parameter. Indeed, using the maximization of the likelihood as a criterium, it is well-known that the Poisson density function parameter over a day can be estimated by the average daily values of the demand over this day [26]. The analytical proof of this result is recalled in Appendix.

The quality of the samples containing the average daily demand values over $\bar{d}$ days, used as input of our learning approaches, is compared in two ways:

- The heterogeneity of the data sample is measured using the Gini index [26]. We recall briefly how it is derived. The input sample $\lambda(j),\ j=1,...,\bar{d}$ is ordered according to increasing values. The resulting

ordered statistic is denoted: $\lambda^{(1)} \leq \lambda^{(2)} \leq ... \leq \lambda^{(\bar{d})}$. The Gini index $\mathcal{G}$ is then defined as follows:

$$\mathcal{G} \equiv \frac{2\sum_{j=1,...,\bar{d}} j\lambda^{(j)}}{\bar{d}\sum_{j=1,...,\bar{d}} \lambda^{(j)}} - \frac{\bar{d}+1}{\bar{d}}$$

- The quantity of information contained in the input data is measured through the Shannon entropy [24], which captures the uncertainty associated with the demand variable realizations. This is a measure of the resulting distribution disparity compared to the uniform density[12] i.e., the information that it brings compared to the uniform density which does not contain any. Practically, we compute it using a histogram of the input variables. Suppose we let C be the optimal number of classes associated with the histogramm issued from the input data. We detail how it is obtained. We fix a precision level $\epsilon > 0$, beween two classes. Using the normalised mean squared error definition (cf. Equation (1)), we can express the precision as a function of the input sample i.e., $\epsilon = \frac{l_j^2}{\frac{1}{\bar{d}^2}(\sum_{j=1,...,\bar{d}} \lambda(j))^2}$, where $l_j \equiv \lambda(j+1) - \lambda(j)$ is the length of class j. This implies in turn that $l_j = \sqrt{\epsilon}\frac{1}{\bar{d}}\sum_{j=1,...,\bar{d}} \lambda(j)$. The optimal number of classes is finally obtained as: $C = \frac{\max_{j=1,...,\bar{d}} \lambda(j) - \min_{j=1,...,\bar{d}} \lambda(j)}{\sqrt{\epsilon}\frac{1}{\bar{d}}\sum_{j=1,...,\bar{d}} \lambda(j)}$. If we let f_i be class i frequency, we obtain the entropy measure associated with the input data sample according to the following equation:

$$\mathcal{E} \equiv \sum_{i=1}^{C} f_i \log \frac{1}{f_i}$$

For the artificial data base issued from the bayesian network, we obtain a Gini index of 0.354 and an estimated entropy value of 0.840 (with a precision of 10^{-3}). For the second data base containing real data about the household electricicty consumption in 2011, we obtain a Gini index of 0.406 and an estimated entropy value of 0.884 (with a precision of 10^{-3}).

These information measure results imply that both data sets contain highly dispersed data but rather low levels of information, increasing the difficulty to build accurate forecasters.

3.4 Measures to compare the learning approach performances

The sharing of the data bases in two sets are performed according to the following way: for the artificial data base, the training set is made of the first 9 months of the generated year demand and the test set contains the three remaining months. For the real data base, the training set contains the first 180 days of year 2011 and the test set, the remaining ones until august 2011.

The learning approaches performances are compared in two ways:

- The accuracy of the approach over the training set is evaluated by plotting the correspondance of the true demand parameter for each day $\lambda(j),\ j = 1, ..., \bar{d}$ over the x- axis versus the estimated one $\hat{\lambda}(j),\ j = 1, ..., \bar{d}$ over the y- axis.
- To compare the non linear time series learning methods performances over the test set, we use the normalized mean squared error (MSE(j)) which computes the square of the error between the true

[12] The uniform density maximizes the entropy.

demand parameter value and the estimated one, normalised over the whole training set. Formally, over day $j = 1, ..., \bar{d}$, it can be described as:

$$\text{MSE}(j) = \Big(\frac{\hat{\lambda}(j) - \lambda(j)}{\frac{1}{|\text{Training set}|} \sum_{j \in \text{Training set}} \lambda(j)} \Big)^2 \tag{1}$$

The criterium needs to be adapted in case we consider regret minimization learning. Indeed, this alternative learning approach produces as output an estimated density function over the set of all the possible demand parameter values: $q_j(.)$ for each day j. Theerfore, in this case, the MSE(j) is adapted to give:

$$\text{MSE}(j) = \frac{1}{\Big(\frac{1}{|\text{Training set}|} \sum_{j \in \text{Training set}} \lambda(j) \Big)^2} \sum_{\lambda \in \Lambda} \Big(\lambda - \lambda(j) \Big)^2 q_j(\lambda) \tag{2}$$

We define now properly the notion of local/expanded accuracy of the learning approaches. For the artificial data base, the test set contains 21 values. Each value coincides with the average daily demand over a day of a specific month. For the test set issued from the real data base, it contains the average daily demand value over 34 days of year 2011 i.e., from $06/28$ to $08/02$. The local accuracy of the learning method is then evaluated as a mean of the first 7 predictions made over the artificial test set and over the first 11 predictions made over the real data test set, whereas the extended accuracy is computed over the remaining predictions made over the test sets.

Finally, the ability to capture uncertainty is measured through the learning approaches capability to predict the true demand parameter when this latter varies a lot i.e., reaches extreme values. To perform this point, we focus on the 7-th day of the real data test base where the true demand parameter reaches its highest mean value.

4 A partial information game

In this section, the set containing all the possible demand parameters is supposed finite i.e., $|\Lambda| < +\infty$[13]. The total number of EVs, on the considered geographic area is denoted $N \in \mathbb{N}$. It is supposed fixed since we restrict the model to a single geographic area and known publicly.

Taking a system point of view, the problem can be envisaged as a game with two players sharing a scarce resource:

- On the one hand, *Nature* which contains the EV drivers, whose behaviors can be influenced by intrinsic preferences, unexpected chance events, and also potentially, offers from rival service providers. Note that it is quite classical in game theory to aggregate atomic behaviors like the EV drivers' ones in a single meta-player [17], [27].
- On the other hand, the *forecaster* who coincides with the service provider, who manages the charge station.
- The underlying *resource* i.e., the charge station, is limited by its capacity to welcome numerous EV drivers. Indeed, it is distributed by a fixed number of charge spots and waiting car parks. Therefore, as usual in game theory, the underlying resource can be considered as scarce [17].

[13] Note however, that it can be arbitrarily large.

The game setting as defined above, is rather unusual in game theory. Indeed, here, Nature is seen as an *oblivious opponent* in the sense that it does not react strategically to the forecaster's actions, but extensions to non oblivious opponents who react strategically, are possible [4].

As already mentioned in Section 1, it is important to note that the electricity demand does not in general, coincide with the number of EV drivers reloading in the service station i.e., the consumption level. Indeed, at time instant t, some EV drivers might decide to delay their reloading to another time instant, or get discouraged at the station because a queue has formed, or be blocked at the station entrance in case where the station capacity limitation has been reached. To be consistent with the reality, we assume in this section that the incoming demand is neither observed nor monitored. In such a partial information framework, the forecaster needs to make predictions about the expected number of EV drivers coming in the station, over each day. To optimize his predictions, the service provider receives daily feedbacks corresponding to his utility. The service provider's utility is a function of the revenue generated by the EV drivers' charges and of the refund that he plans to offers to the EV drivers to compensate for long waiting times/service unavailability. The use of refunds to compensate for dissatisfaction is quite a common pratice in all industries using revenue management techniques [15], [27]. The diagram of the physical/financial flows between the various players is pictured in Figure 5.

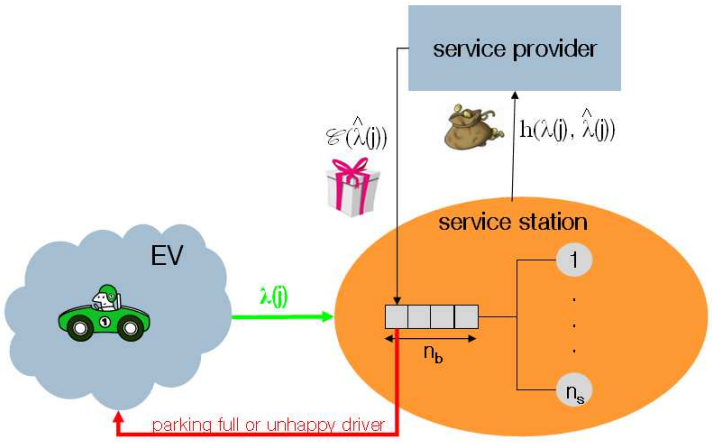

Figure 5: Flow diagram.

4.1 Description of the queue: a scarce resource

We assume that over day $j = 1, ..., \bar{d}$, the EV drivers arriving times are independent and identically distributed (i.i.d.) random variables, distributed according to a Poisson process of parameter $\lambda(j) \in \Lambda$ [2], [26]. We suppose that the considered geographic area is associated with a $M/M/n_s/n_s + n_b$ queue as pictured in Figure 6, where n_s is the number of charge spots in the service station and n_b is the buffer length [2]. Practically, it means that n_s EV drivers can reload their vehicle simultaneously while a maximum number of n_b EV drivers can wait in the surroundings of the station. It is reasonable to assume that $n_s \leq n_s + n_b < N$. The station mean service time can be estimated as $\frac{1}{\mu}$, $\mu \in \mathbb{R}$ being a parameter of the station known publicly.

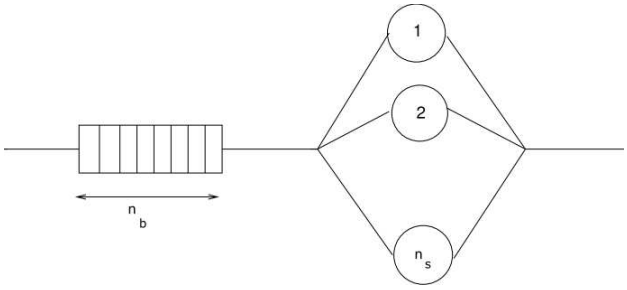

Figure 6: The $M/M/n_s/n_s + n_b$ queue.

To estimate the number of EV drivers in the station at a specific time instant i.e., the consumption level, it is necessary to explicit the probability of the occurence that $0, 1, 2, ..., N$ EV drivers enter the station simultaneously. To make it possible, we assume that the demand is *stationary* over a day. This assumption enables us to obtain the stationary probability distributions over the number of EVs in the station. We recall briefly their analytical expressions below. More precise descriptions of how we derive these stationary probabilities can be found in Appendix and in queueing theory literature [2].

Let $\pi_k(\lambda(j)),\ k = 0, 1, ..., n_s + n_b$ denotes the probability that k EV drivers are in the service station at day j. Note that this probability depends on the mean demand parameter over the considered day $j = 1, ..., \bar{d}$. The probability that $k + 1$ EV drivers enter in the station at a specific time instant takes the form:

$$\pi_{k+1}(\lambda(j)) = \frac{\lambda(j)^{k+1}}{n_s^{k-n_s+1} n_s! \mu^{k+1}} \pi_0(\lambda(j)),\ k = 0, 1, ..., n_s + n_b - 1$$

Besides, the normalization constraint on the probability vector imposes that:

$$\pi_0(\lambda(j)) = \Big[1 + \sum_{k=0,...,n_s-1} \frac{\rho(\lambda(j))^{k+1}}{(k+1)!} + \sum_{k=n_s,...,n_s+n_b-1} \frac{\rho(\lambda(j))^{k+1}}{(n_s^{k-n_s+1}) n_s!}\Big]^{-1}$$

where the charge rate for day j is defined as $\rho(\lambda(j)) \equiv \frac{\lambda(j)}{\mu}$.

Additionally, the queueing approach enables us to characterize the station occupancy. To that purpose, we recall the $M/M/n_s/n_s + n_b$ queue main performance measures [2]:

- The *mean waiting time*: using Little formula [2], the mean waiting time $\mathbb{E}[\tau]$ can be expressed as the ratio of the average number of EV drivers in the station and of the arrival rate. Therefore:

$$\mathbb{E}[\tau] = \frac{1}{\lambda(j)} \Big[\sum_{k=0,...,n_s-1} \frac{k\lambda(j)^{k+1}}{(k+1)! \mu^{k+1}} \pi_0(\lambda(j)) + \sum_{k=n_s,...,n_s+n_b} \frac{k\lambda(j)^{k+1}}{n_s^{k-n_s+1} n_s! \mu^{k+1}} \pi_0(\lambda(j)) \Big]$$

- The *blocking probability* is defined as the probability that the n_s servers and the n_b car parks in the buffer are occupied at the same time. In this case, no more EV drivers can be accepted in the station. It takes the form: $\pi_{n_s+n_b}(\lambda(j)) = \frac{\lambda(j)^{n_s+n_b}}{n_s^{n_b} n_s! \mu^{n_s+n_b}} \pi_0(\lambda(j))$.

Taking into account the case where the station is blocked enables us to incorporate in our model, the possibility that the EV driver might choose another station to reload his car. This other station might belong to the same provider or to a rival one.

4.2 A repeated game with signals

The two player game between Nature and the forecaster is repeated over each day. As already stated, the forecaster receives daily feedbacks coinciding with his utility output. We assume that over each day $j = 1, ..., \overline{d}$, the feedback function takes values in a finite set of signals $\mathcal{S} = \{s_1, ..., s_m\}$ with $m > 0$. As a result, the problem is identified as a *two player repeated game with signals* [1], [17].

Let introduce some notational conventions that will be used to define the service provider's utility. To begin $(x)_+$ designates the non-negative part of real $x \in \mathbb{R}$, $\lfloor x \rfloor$ contains the largest smallest integer below real x. $p \in \mathbb{R}_+$ contains the service provider's price for a complete reload. It is time-invariant and identical for all the EV drivers. $\mathcal{C}(.) : \Lambda \to \mathbb{N}$ represents a unit refund offered to the EV drivers to prevent them from leaving the queue when all the servers are occupied. It is supposed increasing in $\hat{\lambda}(j)$. For the sake of simplicity, we suppose that $\mathcal{C}(\hat{\lambda}(j)) = \lfloor \hat{\lambda}(j) \rfloor, \ \forall j = 1, ..., \overline{d}$.

Over day $j = 1, ..., \overline{d}$, the one-shot game timing can be described as follows:

1. The incoming demand in the geographic area is generated according to a Poisson process of parameter $\lambda(j)$: $D(j, t_l) \sim \mathcal{P}(\lambda(j))$ for $l = 1, ..., n$. It is *hidden* to the forecaster.
2. The forecaster makes a prediction $\hat{\lambda}(j) \in \Lambda$ and sets aside, at the beginning of the day, a refund, that he plans to offer to the EV drivers waiting to reload, in the car parks.
3. The forecaster incurs loss $l(\lambda(j), \hat{\lambda}(j)) = \Big(\lambda(j) - \hat{\lambda}(j)\Big)^2$ and each action $\lambda \in \Lambda$ for the forecaster incurs loss $l(\lambda(j), \lambda) = \Big(\lambda(j) - \lambda\Big)^2$. *None of these values is revealed to the forecaster.*
4. The feedback

$$\begin{aligned} h(\lambda(j), \hat{\lambda}(j)) \quad = \quad & -\underbrace{\mathcal{C}(\hat{\lambda}(j))}_{\text{Refund}} + p\Big[\underbrace{\lfloor \sum_{i=1,...,n} D(j, t_i) \sum_{l=1,...,n_s+n_b} \pi_l(\lambda(j)) \rfloor}_{\text{True expected value of the total number of EVs reloading}} \\ & - \underbrace{(\lfloor \sum_{i=1,...,n} D(j, t_i) \sum_{l=n_s+1,...,n_s+n_b} \pi_l(\lambda(j)) \rfloor - \mathcal{C}(\hat{\lambda}(j)))_+}_{\text{True expected loss of money}} \Big] \end{aligned}$$

   is revealed to the forecaster. The feedback coincides with the service provider's utility. It is the sum of the revenue generated by the mean number of EV drivers who enter the station, minus the positive part of the difference between the mean number of EV drivers waiting to reload in the car parks and the refund that the service provider has planned to attribute at the beginning of the day, minus the cost of this refund.

To solve this two player repeated game with signals, we resort to use an algorithmic approach because analytical approaches seems intractable. To reach this goal, it is useful to store the loss and feedback functions in matrices: the matrix of loss is denoted $\mathfrak{L} = \Big(l(\lambda, \lambda')\Big)_{\lambda, \lambda' \in \Lambda}$. It is known by the forecaster who can compute all the possible combinations of demand parameters. Concerning the feedback matrix, eventhough the true expected value of the total number of EVs reloading cannot be observed before they occur, the forecaster is supposed to know the total number of EVs which have been bought on the market: $N \in \mathbb{N}$. To be consistent with the model, it means that to each value of the demand parameter λ, we can associate an integer number belonging to $[\![0; N]\!]$. As a result, it is possible to store all the combinations of parameter values in a day invariant feedback matrix $\mathcal{H} = (h(\lambda, \lambda'))_{\lambda, \lambda' \in \Lambda}$.

4.3 Resolution of the partial information game

We test two regret criteria to build learning algorithms for the forecaster under partial monitoring: external and internal regret minimizations [4]. Note that other learning rules based on different regret criteria exist such as: regret-matching [10] and regret testing [8]. But, they are rather unefficient judging by the time necessary to reach an equilibrium and offer no guarantee on the reachability of a Nash equilibrium [17].

The prediction problem for the forecaster is considered under randomization. It means that over each day $j = 1, ..., \overline{d}$, the forecaster aims at determining a density function $q_j = (q_j(1), ..., q_j(|\Lambda|))$ over the finite dimension set Λ and chooses his density parameter estimate according to this distribution.

Note that in game theory, the output density function $q_j(.)$, coincides with the forecaster's *strategy* over day j. A strategy can be *mixed* in case where the density function support is not reduced to a single point and *pure* otherwise i.e., in case where it coincides with a Dirac function [17].

We give descriptions about the algorithmic resolutions below.

4.3.1 External regret minimization

The forecaster's regret over the sequence of days $1, ..., \overline{d}$, is expressed as the realized difference between the cumulative loss and the loss of the best demand parameter i.e., pure strategy:

$$\mathfrak{R}^e\Big(\underbrace{\lambda(1), ..., \lambda(\overline{d})}_{\text{true parameter values}}, \quad \underbrace{\hat{\lambda}(1), ..., \hat{\lambda}(\overline{d})}_{\text{estimated parameter values}} \Big) \equiv \sum_{j=1,...,\overline{d}} l(\lambda(j), \hat{\lambda}(j)) - \min_{\lambda \in \Lambda} \sum_{j=1,...,\overline{d}} l(\lambda(j), \lambda) \tag{3}$$

We want to design a randomized algorithm for the forecaster such that his regret $\mathfrak{R}^e\Big(\lambda(1), ..., \lambda(\overline{d}), \hat{\lambda}(1), ..., \hat{\lambda}(\overline{d})\Big)$, is $o(\overline{d})$ regardless of the sequence $\lambda(1),\ \lambda(2), ..., \lambda(\overline{d})$ of demand density parameters. We apply the approach detailed by Cesa-Bianchi and Lugosi in [4].

Algorithm 1: **Randomized prediction under external regret minimization**
For each day $j = 1, 2, ..., \overline{d}$
Parameters: matrix of losses $\mathfrak{L}$, feedback matrix $\mathcal{H}$, matrix $\mathfrak{K}$ such that $\mathfrak{L} = \mathfrak{K}\mathcal{H}$, real numbers $\eta, \gamma \in \mathbb{R}$ such that $0 < \eta, \gamma < 1$.
Initialisation: $w_0 = (1, ..., 1)$.

- Draw an action $\hat{\lambda}(j) \in \Lambda$ according to the distribution:

$$q_j(\lambda) = (1 - \gamma)\frac{w_{j-1}(\lambda)}{W_{j-1}} + \frac{\gamma}{|\Lambda|},\ \forall \lambda \in \Lambda$$

  where $W_{j-1} \equiv \sum_{\lambda \in \Lambda} w_{j-1}(\lambda)$.

- Get feedback $h(\lambda(j), \hat{\lambda}(j))$ and compute $\tilde{l}_j(\lambda) = \frac{\mathfrak{K}(\lambda, \hat{\lambda}(j))h_j}{q_j(\hat{\lambda}(j))},\ \forall \lambda \in \Lambda$.

- Update $w_j(\lambda) = w_{j-1}(\lambda) \exp\Big(-\eta \tilde{l}_j(\lambda)\Big),\ \forall \lambda \in \Lambda$.

4.3.2 Internal regret minimization

By definition, a strategy has a small internal regret if for each couple of parameters $(\lambda, \lambda') \in \Lambda^2$, the forecaster does not regret of not having chosen parameter λ' each day he has chosen parameter λ [4]. We now give a formal definition of this criterium. The internal cumulative regret of strategy $q_j(.)$ is defined by:

$$\max_{\lambda,\lambda' \in \Lambda} \mathfrak{R}^i(\lambda, \lambda', \overline{d}) \quad = \quad \max_{\lambda,\lambda' \in \Lambda} \sum_{j=1,...,\overline{d}} \underbrace{q_j(\lambda)\Big[\tilde{l}_j(\lambda) - \tilde{l}_j(\lambda')\Big]}_{r(\lambda,\lambda',j)} \tag{4}$$

Where $r(\lambda, \lambda', \overline{d})$ expresses the forecaster's expected regret of having taken parameter λ instead of parameter λ'. More precisely, we have the following relations:

$$\begin{aligned} r(\lambda, \lambda', j) &= q_j(\lambda)\Big[\tilde{l}_j(\lambda) - \tilde{l}_j(\lambda')\Big] \\ &= \mathbb{E}[\mathbf{1}_{\hat{\lambda}(j)=\lambda}\Big(\tilde{l}_j(\hat{\lambda}(j)) - \tilde{l}_j(\lambda')\Big)] \end{aligned}$$

$r(\lambda, \lambda', j)$ can be alternatively seen as the forecaster's regret of having put the probability mass $q_j(\lambda)$ on parameter λ instead of on parameter λ'. Using Equations (3) and (4), it is straightforward to check that any algorithm with a small internal regret also has a small external one [4]. We now detail how to proceed to obtain a small internal regret minimizing strategy. We adapt Cesa-Bianchi and Lugosi description to our problem [4] and proceed recursively.

Algorithm 2: **Randomized prediction under internal regret minimization**

- At day $j = 1$, we initialize the forecaster so that $q_1 = (\frac{1}{|\Lambda|}, ..., \frac{1}{|\Lambda|})$ coincides with the uniform distribution over the $|\Lambda|$ possible parameter values.
- Consider day $j > 1$. At day $j-1$, the forecaster has chosen $q_{j-1} = (q_{j-1}(1), ..., q_{j-1}(|\Lambda|))$. For each couple of parameters $\lambda, \lambda' \in \Lambda$ such that $\lambda \neq \lambda'$, we transport the probability mass from λ to λ' giving rise to probability vector $q_{j-1}^{\lambda \to \lambda'}$. The updating process is described as follows:

$$\begin{cases} q_{j-1}^{\lambda \to \lambda'}(k) = q_{j-1}(k), \ \forall k = 1, ..., |\Lambda|, \ k \neq \lambda, \ k \neq \lambda' \\ q_{j-1}^{\lambda \to \lambda'}(\lambda) = 0 \\ q_{j-1}^{\lambda \to \lambda'}(\lambda') = q_{j-1}(\lambda) + q_{j-1}(\lambda') \end{cases}$$

  Let $\Delta(., ., j)$ be the probability distribution over the couples of parameter $\lambda \neq \lambda'$ such that:

$$\frac{1}{\overline{d}} \sum_{j=1,...,\overline{d}} \sum_{\lambda \neq \lambda'} \tilde{l}^E(q_j^{\lambda \to \lambda'}, \lambda(j)) \Delta(\lambda, \lambda', j) \leq \min_{s \neq s'} \frac{1}{\overline{d}} \sum_{j=1,...,\overline{d}} \tilde{l}^E(q_j^{s \to s'}, \lambda(j)) + \varepsilon_{\overline{d}}$$

  where $\varepsilon_{\overline{d}} \to 0$ and the expected loss associated with strategy $q_j(.)$ is defined as: $\tilde{l}^E(q_j, \lambda(j)) = \sum_{\lambda \in \Lambda} \tilde{l}_j(\lambda) q_j(\lambda)$.
- We choose the exponentially weighted average strategy [4] over all the couples of parameters:

$$\Delta(\lambda, \lambda', j) = \frac{\exp(-\eta \sum_{i=1,...,j-1} \tilde{l}^E(q_i^{\lambda \to \lambda'}, \lambda(i)))}{\sum_{s \neq s'} \exp(-\eta \sum_{i=1,...,j-1} \tilde{l}^E(q_i^{s \to s'}, \lambda(i)))}$$

- Over day j, the updated strategy is obtained as solution of the fixed point equation:

$$q_j = \sum_{\lambda \neq \lambda'} q_j^{\lambda \to \lambda'} \Delta(\lambda, \lambda', j)$$

This equation can be simplified to take a matricial form:

$$(\sum_{\lambda \neq \lambda'} \Phi_{\lambda,\lambda',j} - I_{|\Lambda|,|\Lambda|}) = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} \tag{5}$$

where $I_{|\Lambda|,|\Lambda|}$ is the $|\Lambda| \times |\Lambda|$- dimensional identity matrix, the right part of Equation (5) contains a $|\Lambda|$- dimensional zero vector. For every couple of parameters $\lambda, \lambda' \in \Lambda$ such that $\lambda \neq \lambda'$, matrix $\Phi_{\lambda,\lambda',j}$ coefficients are null except on its diagonal and on the λ'- th row:

$$\begin{aligned} \Phi_{\lambda,\lambda',j}(k,k') &= 0 \text{ for } k,k' \neq \lambda, k,k' \neq \lambda', k \neq k' \\ \Phi_{\lambda,\lambda',j}(\lambda,\lambda) &= 0 \\ \Phi_{\lambda,\lambda',j}(\lambda',\lambda) &= \Delta(\lambda,\lambda',j) \text{ and } \Phi_{\lambda,\lambda',j}(\lambda',\lambda') = \Delta(\lambda,\lambda',j) \end{aligned}$$

4.3.3 Stability of the regret minimization algorithms over artificial and real data bases

We start by applying the external regret minimization learning as described in Algorithm 1, to the data bases described in Section 3. The learning approach performances are evaluated using the measure criteria defined in Section 3.4, using probabilistic MSE as defined in Equation (2). We observe in Figures 7 and 8 that the accuracy of the learning stage over the training set is very bad. It is as if the forecaster chooses to explore all the possible parameter values to better exploit the acquired information over the test set. Indeed, the external regret minimization performances seems quite promising especially for the real data (cf. Figures 9 and 10).

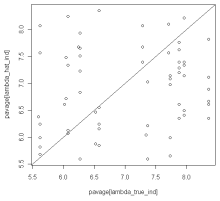

Figure 7: Accuracy of the training under external regret minimization over artificial data.

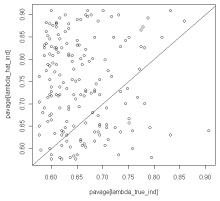

Figure 8: Accuracy of the training under external regret minimization over real data.

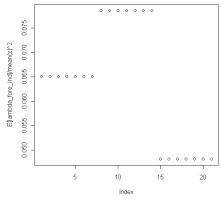

Figure 9: MSE under external regret minimization over artificial data.

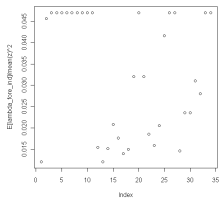

Figure 10: MSE under external regret minimization over real data.

In a second experiment, we apply the internal regret minimization learning as described in Algorithm 2, to the data bases described in Section 3. We observe that as under external regret minimization, the accuracy of the learning stage over the training set seems not very good (cf. Figures 11 and 12). It might be that the forecaster uses this stage to explore all the possible parameter values and exploit the resulting information

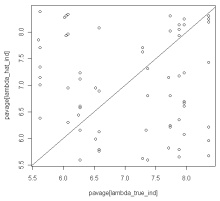

Figure 11: Accuracy of the training over artificial data.

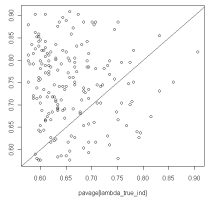

Figure 12: Accuracy of the training over real data.

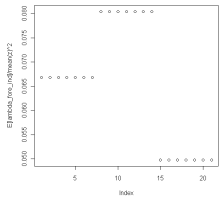

Figure 13: MSE under internal regret minimization over artificial data.

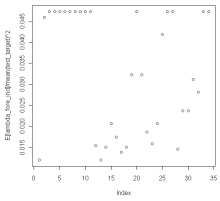

Figure 14: MSE under internal regret minimization over real data.

to optimize his predictions over the test set. The performances of the internal regret minimization algorithm over the test set are rather similar with the external regret minimization algorithm (cf. Figures 13 and 14).

We recall that in our game setting, a Nash equilibrium is reached when no unilateral deviation from the forecaster (respectively, Nature)'s strategy at equilibrium can decrease the forecaster's loss (respectively, increases Nature's payoff). Here, Nature's payoff coincides with the sum of all the EV drivers' atomic utilities. In the considered two player repeated game with signals, the Nash equilibrium set is non null, since the game is finite (two players, action set $|\Lambda| < +\infty$) and therefore it admits at least one Nash equilibrium [17]. This Nash equilibrium might be approximated as the convergence point of some well-designed learning strategies using algorithmic game theory [18]. However, these learning strategies might be very difficult to build or the time necessary to ensure their convergence might be over-sized.

As a result, far wider notions of equilibrium have been introduced to study the stability of games with signals: a correlated equilibrium is a probability distribution on the couple of demand parameters (both true and estimated), which can be interpreted as the distribution of play instructions given to the players by some regulator/referee [1]. Each player (i.e., forecaster and Nature) is given privately instructions for his own play only. Also, for every possible instruction that a player receives, the player realizes that the instruction provides a best response to the random estimated play of the other, assuming they both follow their instructions. In general the received signals are correlated, but if they are independent, the correlated equilibrium is a Nash equilibrium (either pure or mixed). As a result, the Nash equilibrium set is included in the correlated equilibrium.

We recall below two probability notions that will be useful in the rest of the section, to characterize the convergence of our learning algorithms.

- The *marginal empirical frequency of play* for the forecaster over day j is defined as: $q_j^e(\lambda) = \frac{1}{j}\sum_{s=1,\dots,j} \mathbf{1}_{\hat{\lambda}(s)=\lambda}$ and for Nature, it takes the form: $p_j^e(\lambda) = \frac{1}{j}\sum_{s=1,\dots,j} \mathbf{1}_{\lambda(s)=\lambda}$.
- The *joint empirical frequencies of play* is defined as:

$$P_j^e(\lambda, \lambda') = \frac{1}{j}\sum_{s=1}^{j} \mathbf{1}_{\hat{\lambda}(s)=\lambda, \lambda(s)=\lambda'}$$

As an output, it is possible to define the product of the marginal empirical frequencies of play: $q_j^e \times p_j^e(\lambda) = \frac{1}{j^2}(\sum_{s=1}^{j} \mathbf{1}_{\hat{\lambda}(s)=\lambda})(\sum_{s=1}^{j} \mathbf{1}_{\lambda(s)=\lambda})$. According to Cesa-Bianchi and Lugosi, if both players play according to a strategy minimizing internal regret, the joint empirical frequencies of play converges to a

correlated equilibrium [4]. However, in this article, Nature being oblivious, we have no guarantees to be in the worst case.

We have plotted the forecaster's joint empirical frequencies of play over the $|\Lambda| \times |\Lambda|$ plane, at various days, starting down right from the farthest date and finishing to day $\bar{d}$ at the top right. We do not observe any convergence for the artificial data set since the plots of the joint empirical frequencies of play at day $\bar{d}$ is quite distinct from the one at day $\bar{d}-1$. This observation can be easily explained using two arguments. First, Nature is oblivious i.e., does not necessarily follows an internal regret minimization strategy. Second, in the bayesian network, the data were generated using a gaussian density function whereas the forecaster tries to predict the demand using a Poisson process. Poisson density can be seen as a discretized approximation of the gaussian density only when the gaussian density mean and variance coincide which might not be the case here. Additionally, the size of the test set might be insufficient to reach an equilibrium for both players' strategies. On the contrary, convergence of the joint empirical frequencies of play is observed over the real household electricity consumption data set. Furthermore, the joint empricial frequency of play puts its heighest weights in two points which correspond to the case where the forecaster makes biased forecasts of the true demand parameter. It implies that it seems to be in the interest of the forecaster to provide biased estimates of the true demand parameter over the training set to optimize the internal regret criterium.

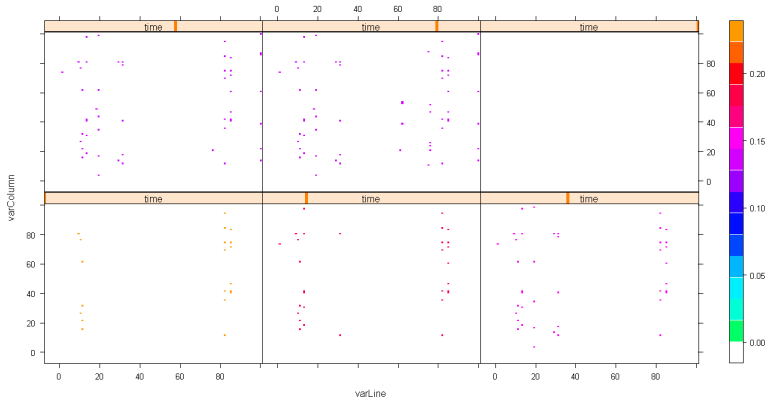

Figure 15: Convergence of the joint empirical frequencies of play over artificial data.

We have also plotted the product of the marginal empirical frequencies of play under external regret minimization strategy. The results over both data sets are identical to the ones obtained in Figures 15 and 16. This implies that over the real data set, the joint empirical frequency of play coincides with the product of the marginal empirical frequencies of play i.e., at equilibrium the received signals are independent. Therefore, the resulting strategies for the forecaster and Nature form a correlated equilibrium which is also a Nash equilibrium. Therefore, the forecaster should have no incentive to deviate unilateraly from his strategy. Note however that it does not constitute a proof of the unicity of the Nash equilibrium.

To sum up, the main interesting aspects of regret minimization based approaches can be summarized below:

- They do not require the monitoring of the incoming demand but only his feedback monitoring.
- They are adaptative, therefore more capable to capture the uncertainty or unexpected values in the

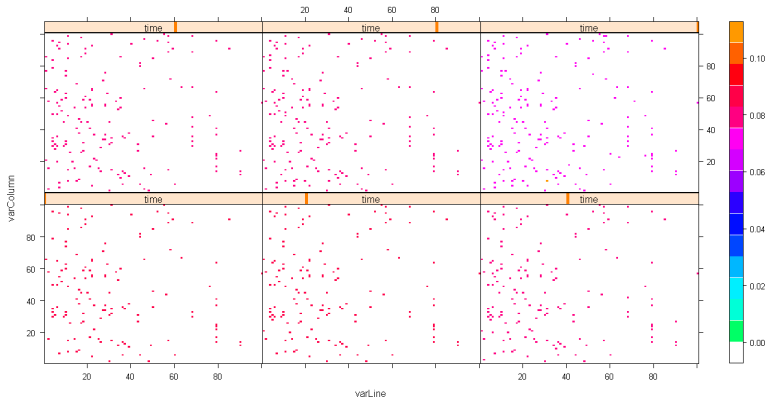

Figure 16: Convergence of the joint empirical frequencies of play over real data.

demand values, like extreme values.

- They are more scalable to competition. Indeed, they might take into account one's adversary's reaction. Here, the adversary has been associated with Nature, but it can be generalized to another station sharing the same geographic area and managed by the same provider or even, by a rival.
- In all generalities, they do not require the use of a stochastic model for the generation of the partially observed phenomena, although we choose to make this assumption in the present article.
- They enable the solving of both full (i.e., bandit based) and partial information problems.
- They might provide guarantees on the resulting strategy by avoiding dominated ones [4], [17], and converging under internal regret minimization, to correlated equilibria [4].

5 Comparison of regret minimization learning with non linear time series models

In this section, we introduce briefly and test on the data bases presented in Section 3, three well-known learning approaches: Neural Network (NN), Support Vector Machine (SVM), AutoRegressive Integrated Moving Average process (ARIMA). The various learning approaches are finally ranked according to their performances, in Section 5.4.

5.1 Neural Network approach (NN)

There are many articles dealing with Neural Network (NN) applications to electricity consumption forecasting on short, medium and long-time scales [22], [25], [30], weather (especially precipitations) [9], and financial forecasting [13]. It has become an attractive tool due to several distinguishing characteristics: first, NN is capable of modelling complex relationships between input and output patterns. It learns from example and may capture complex relationships without any theoretical understanding of the underlying system.

Second, as reported in the literature, NN is often able to infer the unseen parts of the data correctly even if the sample data contains noisy information.

Yann and Choo compare the performances of NN versus classical time series forecasting methods to predict long-term electricity consumption in Malaysia [30]. They conclude that errors for NN models are smaller than in the classical time series based approaches. However, as noted by Lendasse et al. [14], the common difficulty of NN application is the determination of sufficient and necessary information to learn accurate predictive models. If information is insufficient, predictions will be poor. If, information is useless or redundant, modeling will be difficult or even skewed. Other non linear approaches exist such as self-organizing maps of Kohonen [14], but are out of the scope of the present article.

Briefly, a NN consists of a set of interconnected units, called neurons. The activation state of each neuron is determined by a function depending on the states of the previous neurons. Each pair of neurons is connected by a synapsis, characterized by a directed weighted edge. In the application, we consider feed-forward perceptrons which are organized in ordered layers whose neurons only receive information from the previous layer. A more complete description can be found in [28].

Learning using NN over artificial and real data bases: The numerical illustrations in this section, are performed using R software [21]. The used NN is made of three layers: the input layer, the hidden layer made of 20 interconnected neurons and the output layer. Note that the optimal number of neurons has been determined by experimental set up. The input variables are the day, the month and the demand as learned over the training set. One neuron is associated with each input variable. Note that it is necessary to normalize the input variables. The demand is output over each day. One neuron is associated with the output demand. The prediction is then realized on the mean of the demand over each day.

To evaluate the accuracy of the learning stage over the training set, we have plotted the averaged values of demand over each day $\lambda(j),\ j = 1, ..., \bar{d}$ versus the estimated ones $\hat{\lambda}(j),\ j = 1, ..., \bar{d}$ in Figure 17 for the artificial data and in Figure 18 for the real ones. We observe that the accuracy is rather good in both cases. Then, we have computed the normalised MSE as defined in Equation (1), for the artificial data (Figure 19) and for the real ones (Figure 20) over the test set, comparing the true average demand to the estimated one. The quality of the forecast appears rather good in the long-term for the artificial data but rather poor for the real data on local and intermediate-terms.

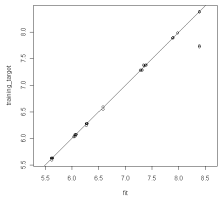

Figure 17: Accuracy of the training over artificial data.

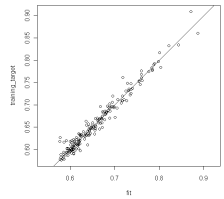

Figure 18: Accuracy of the training over real data.

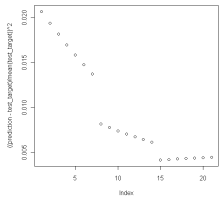

Figure 19: MSE for the NN over artificial data.

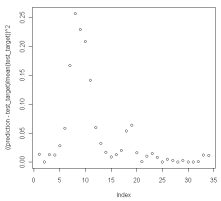

Figure 20: MSE for the NN over real data.

5.2 Support Vector Machine approach (SVM)

Support vector machine approaches (SVM) are traditionally used to perform classification [5]. We use them in two ways: first to perform classification coupled with hierarchical clustering, second to perform

regression. This latter method is called support vector regression (SVR) [6].

Using the first approach, we perform hierarchical clustering of both artificial and real data. It enables us to classify the training data set in a dendrogram (demand distance tree). Then we can infer the optimal number of classes and get the corresponding data from the branches in the tree. For the artificial data, we identify 2 classes which correspond to the cutting of the resulting tree at its highest jump (cf. Figure 21). Whereas, for the real data, we obtain 3 classes (cf. Figure 24).

Note that the tree issued from hierarchical clustering has been built using the euclidian distance as a metric and the Ward's criterium for the aggregation [21].

We then run the SVM algorithm with a radial kernel over the training set to learn the relationship between the demand class and the day and month. Then, we check with the same data, if the class associated by the SVM algorithm to the variables day and month is the same. We describe the relationship between the output of the SVM algorithm and the demand. It depends whether we fall in the binary case or in the ternary case. To value, we associate the centroid of the class to which it belongs. The association rule is described below:

- In the binary case, the demand is estimated as: $\hat{\lambda} = (\lambda_{d,m} - 1)(c_2 - c_1) + c_1$ where $c_k,\ k = 1, 2$ coincides with the center of class k and $\lambda_{d,m}$ is the class given by the SVM algorithm for day d and month m.
- In the ternary case, the demand is estimated as: $\hat{\lambda} = (\lambda_{d,m} - 2)_+(c_3 - c_2) + (\lambda_{c_k} - 2)_-(c_1 - c_2) + c_2$.

Finally, forecasts are performed over the test base and the predictions are compared with the true averaged demand values using the normalised MSE described in Equation (1) as performance criterion.

Learning using classification based SVM over artificial and real data bases: We have plotted the averaged demand values for each day $\lambda(j),\ j = 1, ..., \bar{d}$ versus the estimated ones $\hat{\lambda}(j),\ j = 1, ..., \bar{d}$ in Figure 22 for the artificial data and in Figure 25 for the real data. We observe that for both data bases, the accuracy of the method over the training set seems not convincing. Fortunately, the predictions over the test data base are good according to the MSE criterion, especially in the long-run.

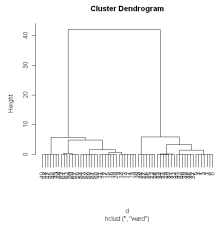

Figure 21: Hierarchical clustering of the artificial data.

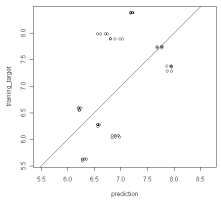

Figure 22: Accuracy of the SVM based on classification over the artificial training set.

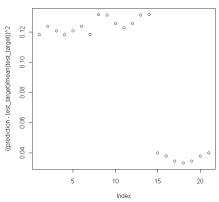

Figure 23: MSE of the SVM based on classification over the artificial test set.

Learning using regression based SVM over artificial and real data bases: Like in the classification based SVM, the accuracy of the learning stage over the training set is not convincing (cf. Figures 27 and 28). But, the prediction stage over the training set is very good, especially in the long-run (cf. Figures 29 and 30), and better than in the classification based SVM approach.

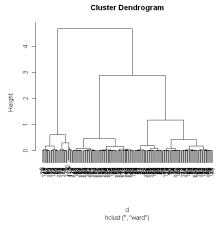

Figure 24: Hierarchical clustering of the real data.

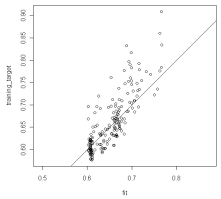

Figure 25: Accuracy of the SVM based on classification over the real data training set.

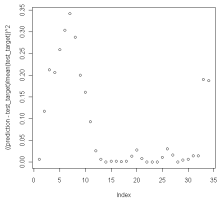

Figure 26: MSE of the SVM based on classification over the real data test set.

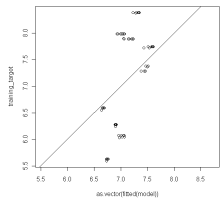

Figure 27: Accuracy of the regression based SVM over artificial training set.

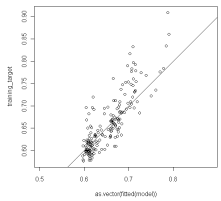

Figure 28: Accuracy of the regression based SVM over the real data training set.

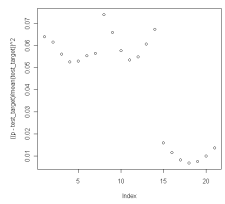

Figure 29: MSE of the regression based SVM over the artificial data test set.

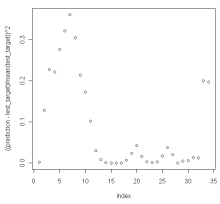

Figure 30: MSE of the regression based SVM over the real data test set.

5.3 AutoRegressive Integrated Moving Average process (ARIMA)

In this section, we consider autoregressive models, which are classical time series analysis techniques, traditionally used for prediction purposes. Autoregressive models are linear Markovian models based on the assumption that the state of the targeted variable can be computed according to a linear combination of its previous states.

An autoregressive integrated moving average (ARIMA) model is a generalization of an autoregressive moving average (ARMA) model. These models are fitted to time series data either to better understand the data or to predict future points in the series [3], [11]. They are applied in some cases where data show evidence of non-stationarity. The model used in this article is generally referred as an ARIMA(p, q) where p and q are non-negative integers that refer to the order of the autoregressive and moving parts of the model respectively. Practically, we apply the ARIMA model with two approaches.

In the first approach, the parameters p, q are determined using a correlogram. Then, the accuracy of the model is computed using the remaining residuals. Forecasts are finally performed on the test sets. All these steps are performed automatically using the R software [21]. During the data analysis, we observe two points:

- The variance of the process decreases towards zero when time step increases.
- The fact that we consider solely the mean of the demand over each day enables us to suppress the seasonal component of the process and as a by-product, to reduce the resulting MSE.

In the second approach, we use a regression based ARIMA of parameters ϕ, μ, β, γ. In this case, the ARIMA is built upon three temporal series: the day X_t, the month Y_t and the averaged demand Z_t. The interest of this approach is to take into account the correlations between the three processes. Formally, demand can be expressed as function of the regressors i.e., day and month, as follows:

$$Z_t - \phi Z_{t-1} = \mu + \beta(X_t - \phi X_{t-1}) + \gamma(Y_t - \phi Y_{t-1}) + \theta\varepsilon_{t-1}$$

where ε_{t-1} is a white noise i.e., it is distributed according to a normal density function centered in 0 and of variance 1 [26]. In this case, the aim of the learning process is to fit the parameters ϕ, μ, β, γ in order to get a simple predictive model. Note that this is a Markovian process of order one i.e., the state of each variable at time instant t depends upon its value at time instant $t-1$ solely.

Learning using ARIMA over artificial and real data bases: We observe in Figures 31 and 32, that the accuracy of the learning stage over the training base seems not convincing. Furthermore, the performances of the prediction stage are better on the overall, for real data than for artificial ones but quite heterogeneous over the training set.

Learning using regression based ARIMA over artificial and real data bases: For regression based ARIMA, the performances of the forecast over the test set are better than using a classical ARIMA model for the artificial data base (cf. Figure 37), but not always for the real data base (cf. Figure 38). Furthermore, the method reaches its best performances for intermediate-term forecasting.

5.4 Performance based ranking of the learning approaches

In this section, we compare the four learning approaches described in Sections 4 and 5, on the two data bases described in Section 3. As explained in Section 3.3, both data bases have been divided into two sets:

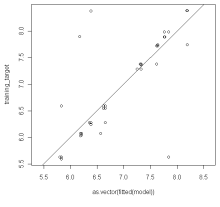

Figure 31: Accuracy of the ARIMA over the artificial data training set.

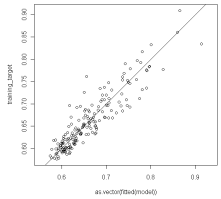

Figure 32: Accuracy of the ARIMA over the real data training set.

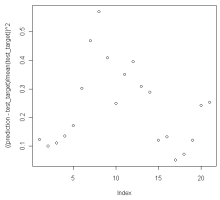

Figure 33: MSE for the ARIMA over artificial data test set.

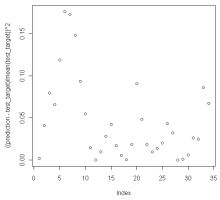

Figure 34: MSE for the ARIMA over the real data test set.

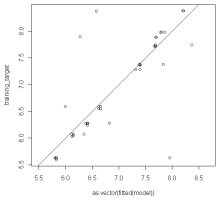

Figure 35: Accuracy of the regression based ARIMA over the artificial data training set.

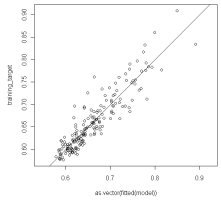

Figure 36: Accuracy of the regression based ARIM over the real data training set.

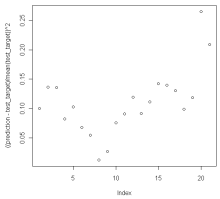

Figure 37: MSE for the regression based ARIMA over the artificial data test set.

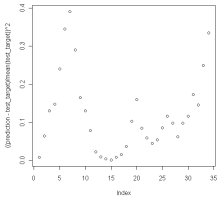

Figure 38: MSE for the regression based ARIMA over the real data test set.

Method / Criterion	Average local accuracy	Average expanded accuracy
NN	1	1
SVM based classification	7	3
SVM based regression	3	2
ARIMA	6	5
ARIMA based regression	2	6
External regret	4	4
Internal regret	5	4

Table 2: Performance based ranking of the learning approaches for the artificial data.

Method / Criterion	Average local accuracy	Average expanded accuracy	Uncertainty handling
NN	3	2	3
SVM based classification	4	4	4
SVM based regression	5	5	5
ARIMA	2	3	2
ARIMA based regression	6	6	6
External regret	1	1	1
Internal regret	1	1	1

Table 3: Performance based ranking of the learning approaches for the real household electricity demand data.

a training set where learning is performed and a test set where the performances of the learning approaches are tested. Then, the performances of the learning methods are evaluated by averaging locally or all over the test set, the MSE, as described in Section 3.4. The capability to handle uncertainty for the learning methods is also evaluated by comparison of the MSE value in an extremal demand value belonging to the real data test set. More precisions have already been given in Section 3.4. Finally, each method is ranked from 1 to 5, the rank increasing as the averaged MSE increases i.e., the smallest the rank, the best the performance. The resulting ranking is given in Table 2 for the artificial data base and in Table 3 for the real data base. Note that rank coincidence corresponds to cases where the averaged MSEs are identical on the local/expanded-term periods. Of course, it does not mean that the MSEs coincide on every days of the local/expanded-term periods.

In Table 2, we observe that NN performs better than the other learning approaches. This can be explained by the fact that the Poisson density function might not be a good approximate of the gaussian density function used to generate the artificial data and also that the training set issued from the artificial data is too restrained. These two points explain why the regret minimization algorithms have not succeeded reaching an equilibrium over the artificial data sets. In Table 3, we conclude that the regret minimization algorithms clearly outperform other learning methods on real data and guarantee a better treatment of the uncertainty resulting from extreme/unexpected events. Besides, as highlighted in Section 4.3.3, a Nash equilibrium is reached after the spanning of the real test set.

6 Conclusion

We have compared the local/expanded accuracy and ability to handle uncertainty of four categories of learning methods: (i) NN, (ii) SVM, (iii) ARIMA process, (iv) Regret minimization algorithms, using as benchmarks two data bases: an artificial one generated using a bayesian network and real domestic household electricity consumption data in southern California. We observe that over real data, regret minimization methods clearly outperform the other learning approaches.

The opportunity to take into account the opponent's selfish maximizing reaction as it is the case in regret based algorithms, should enable us to tackle the geographic dimension of the electricity demand prediction problem and to incorporate competition between the providers' stations.

Cooperation problems might also arise for the EV charging coordination. Indeed, coordination is necessary to ensure that capacity constraints are not exceeded. It requires the design of incentivised mechanisms in order to guess the EV drivers' reservation prices or to force them to not cheat about their arrival, departure and maximum charging speeds [23]. Another interesting aspect concerns the study of the forces inciting on the one hand, the provider to cooperate in order to gather more information, which improves his learning method accuracy and, on the other hand, to bias (voluntariliy or unvoluntariliy) the received information i.e., to deviate from the alliance initial commitment. Punishment mechanisms or at least credible threats should then be implemented to ensure the quality of information communication mechanisms.

Appendix

Why the average daily demand provides good estimates of the generating Poisson density function parameter

Using as input the data base, we estimate the Poisson process parameter over day j by averaging the demand values over the day of interest. We shortly justify why it provides a good estimate of the generating Poisson process parameter. This result is quite classical in parametric statistics estimation theory [26] but we recall a short proof of it below. The mean of the sample containing incoming demand over day j coincides with the estimate of the generating Poisson process, under maximum likelihood criterium. Suppose the electricity demand measures over a day j form a sample of i.i.d. realizations of a random variable distributed according to a Poisson process of unknown parameter $\lambda(j)$. We estimate the electricity demand variable parameter over day j according to the maximum of likelihood criterium. The likelihood associated with sample $(D(j,t_1),...,D(j,t_n))$ over day j is of the form: $\mathcal{L}\Big(j,t_1,...,t_n,\lambda(j)\Big) = \prod_{l=1,...,n} \frac{\lambda(j)^{D(j,t_l)}\exp(-\lambda(j))}{D(j,t_l)!}$. The demand parameter maximizing day j demand sample is called maximum likelihood (ML) estimator. It is obtained as solution of the following optimization problem:

$$\begin{aligned}\lambda(j)^{ML} &= \arg\max_{\lambda(j)\in\Lambda} \mathcal{L}\Big(j,t_1,...,t_n,\lambda(j)\Big)\\ &= \arg\max_{\lambda(j)\in\Lambda} \log\mathcal{L}\Big(j,t_1,...,t_n,\lambda(j)\Big) \text{ the logarithm being bijective}\end{aligned}$$

Differentiating the likelihood with respect to parameter $\lambda(j)$, we infer the analytical expression of the maximum likelihood demand estimate over day j:

$$\lambda(j)^{ML} = \frac{1}{n}\sum_{l=1,...,n} D(j,t_l)$$

Derivation of the stationary probabilities associated with the consumption level

We recall that $\pi_k(\lambda(j)),\ k = 0, 1, ..., n_s + n_b$ denotes the probability that k EV drivers are in the service station at day j. The arriving/departure process in the station can be represented as a Markov chain with discret time discrete space, as illustrated in Figure 39. The states coincide with the number of EV who simultaneously enter the station in order to reload their battery. Due to the capacity limitation in the buffer size/number of servers, the state space remains finite.

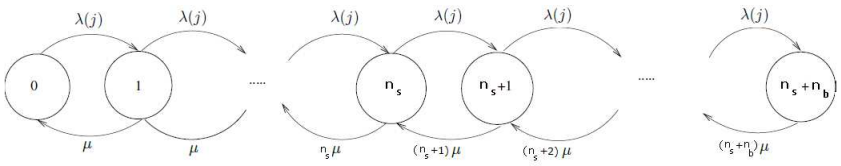

Figure 39: The discrete time discrete space Markov chain associated with the $M/M/n_s/n_s + n_b$ queue.

For $k = 0, 1, ..., n_s - 1$, we have the relation: $\lambda(j)\pi_k(\lambda(j)) = (k+1)\mu\pi_{k+1}(\lambda(j))$, using the Markov chain transition probability definition. Then, by recurrence, we infer the updated relation:

$$\pi_{k+1}(\lambda(j)) = \frac{\lambda(j)^{k+1}}{(k+1)!\mu^{k+1}}\pi_0(\lambda(j)),\ \forall k = 0, 1, ..., n_s - 1 \tag{6}$$

For $k = n_s, ..., n_s + n_b - 1$, we have the relation: $\lambda(j)\pi_k(\lambda(j)) = n_s\mu\pi_{k+1}(\lambda(j)),\ k = n_s, ..., n_s + n_b - 1$. By recurrence, we infer the updated relation:

$$\pi_{k+1}(\lambda(j)) = (\frac{\lambda(j)}{n_s\mu})^{k-n_s+1}\pi_{n_s}(\lambda(j)),\ \forall k = n_s, ..., n_s + n_b - 1 \tag{7}$$

We now substitute the expression of $\pi_{n_s}(\lambda(j))$ as obtained through Equation (6) in Equation (7), which takes the form:

$$\pi_{k+1}(\lambda(j)) = \frac{\lambda(j)^{k+1}}{n_s^{k-n_s+1}n_s!\mu^{k+1}}\pi_0(\lambda(j)) \tag{8}$$

Finally, the normalization constraint on the probability vector $\pi(\lambda(j)) = \Big(\pi_0(\lambda(j)), \pi_1(\lambda(j)), ..., \pi_{n_s+n_b}(\lambda(j))\Big)$ i.e., $\sum_{k=0,...,n_s+n_b}\pi_k(\lambda(j)) = 1$, enables us to infer the analytical expression of probability $\pi_0(\lambda(j))$:

$$\pi_0(\lambda(j)) = \Big[1 + \sum_{k=0,...,n_s-1}\frac{\rho(\lambda(j))^{k+1}}{(k+1)!} + \sum_{k=n_s,...,n_s+n_b-1}\frac{\rho(\lambda(j))^{k+1}}{(n_s^{k-n_s+1})n_s!}\Big]^{-1}$$

where the charge rate for day j is defined as $\rho(\lambda(j)) \equiv \frac{\lambda(j)}{\mu}$.

Acknowledgement. This research has been funded by the ELVIRE project. ELVIRE is a Collaborative Project (STREP) supported by the European Commission in the 7-th Framework Programme (Project Reference: ICT-2007-249105).

Biographies

Hélène Le Cadre graduated from university of Rennes 1/ENS Cachan, France. She holds a Magistère of mathematics and is Télécom Bretagne engineer. She received her Ph.D. in applied mathematics, in 2009, from university of Rennes 1. She works presently as a temporary researcher at CEA (Alternative Energies and Atomic Energy Commission). Her research interests circle around algorithmic game theory, operations research and statistical learning, as well as around applications of these approaches to energy markets, telecommunications business and information economics.

Roman Potarusov graduated from the University of Artois, Béthune, France, in 2008 with Ph.D. degree in Informatics and Automatics. Then he worked for the CEA (Alternative Energies and Atomic Energy Commission) as a Research Engineer. Now he works as a Software Engineer for the University of Valenciennes, Valenciennes, France. His current research interests include combinatorial optimization problems, genetic algorithms and neural networks.

Cédric Auliac received the M.Sc. degree in Mathematics and informatics applied to Biology from University of Evry, France, in 2004 and the Ph.D. in Computational Biology from the Life Sciences Division of the Alternative Energies and Atomic Energy Commission (CEA) in Evry, France, in 2008. After a post-doctoral position he became a research engineer in CEA's Technological Research Division in Saclay, France, in 2010. His current research activities cover data mining and machine learning applications in e-mobility and energy management.

References

[1] Aumann R. J., Correlated equilibrium as an expression of Bayesian rationality, Econometrica, vol.55, pp.1–18, 1987

[2] Baynat B., Queueing theory: From Markov chains to product form networks, Hermes Sciences Publications, 2000

[3] Brockwell P. J., Davis R. A., Introduction to Time Series and Forecasting, Springer, 1996

[4] Cesa-Bianchi N., Lugosi G., Prediction, Learning, and Games, Cambridge University Press, 2006

[5] Cristianini N., Shawe-Taylor J., An Introduction to Support Vector Machines and other kernel-based learning methods, Cambridge University Press, 2006

[6] Drucker H., Burges C. J. C., Kaufman L., Smola A., Vapnik V., Support Vector Regression Machines, Advances in Neural Information Processing Systems, NIPS, pp.155–161, MIT Press

[7] Fortuin L., Five Popular Probability Density Functions: A Comparison in the Field of Stock-Control Models, The Journal of Operational Research Society, vol.31, pp.937–942, 1980

[8] Foster D. P., Peyton H., Learning, Hypothesis testing, and Nash equilibrium, Games and Economic Behavior, Elsevier, vol.45, pp.73–96, 2003

[9] Hall T., Brooks H. E., Doswell C. A., Precipitation Forecasting using a Neural Network, Weather and Forecasting, vol.14, pp.338–345, 1999

[10] Hart S., Mas-Colell A., A Simple Adaptative Procedure Leading to Correlated Equilibrium, Econometrica, vol.68, pp.1127–1150, 2000

[11] Harvey A. C., Time Series Models, 2- nd Edition, Harvester Wheatsheaf, 1993

[12] Jehan G., Problematics issued form elctric vehicle charging: charging infrastructure and energy selling, EDF presentation at CRE forum

[13] Lam M., Neural network techniques for financial performance prediction: integrating fundamental and technical analysis, journal Decision Support System, Special issue: Data mining for financial decision making, vol.37, 2004

[14] Lendasse A., Lee J., Wertz V., Verleysen M., Time Series Forecasting using CCA and Kohonene Maps - Applications to Electricity Consumption, in proc. ESANN, 2000

[15] McGill J. I., Van Ryzin G. J., Revenue Management: Research Overview and Prospects, Transportation Science, vol.33, pp.233–256, 1999

[16] Mitra D., Wang Q., Stochastic Traffic Engineering for Demand Uncertainty and Risk-Aware Network Revenue Management, IEE/ACM Transactions on Networking, vol.13, pp.221–233, 2005

[17] Myerson R., Game theory: Analysis of Conflict, Harvard University Press, 1997

[18] Nisan N., Roughgarden T., Tardos E., Algorithmic Game Theory, Cambridge University Press, 2007

[19] Nora D., The Green Gold Pioneers, Editor Grasset, 2010

[20] Norris J., Markov Chains, Cambridge University Press, 2004

[21] The R project for Statistical Computing, http://www.r-project.org/

[22] Ringwood J. V., Bofelli D., Murray F. T., Forecasting Electricity Demand on Short, Medium and Long Time Scales Using Neural Networks, journal of Intelligent and Robotic Systems, vol.31, pp.129–147, 2001

[23] Robu V., Stein S., Gerding E. H., Parkes D. C., Rogers A., Jennings N. R., An Online Mechanism for Multi-Speed Electric Vehicle Carging, in proc. of 2-nd International Conference on Auctions, Market Mechanisms and Their Applications, 2011

[24] Rubinstein R., Kroese D. P., The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning, Springer-Verlag, 2004

[25] Srinivasan D., Energy demand prediction using GMDH networks, Machine Learning for Signal Processing, vol.72, pp.625–629, 2008

[26] Tassi P., Statistical Methods, 3-rd Edition, Economica, 2004

[27] Tuffin B., Le Cadre H., Bouhtou M., Optimal Pricing strategy with compensation when QoS is not reached, Annals of Telecommunications, vol.65, pp.163–170, 2010

[28] Tim Jones M., Artificial Intelligence: A Systems Approach, Jones & Bartlett Learning, 2008

[29] Wagner H., Principles of Operations Research, Prentice-Hall, Englewood Cliffs, NJ, 1962

[30] Yan L. M., Choon O. H., Neural networks forecasting on electricity consumption in Malaysia, in proc. of the 2-nd IMT-GT Regional Conference on Mathematics, Statistics and Their Applications, 2006